الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التكوين والتعليم المهنيين
Ministère de la Formation et de l'Enseignement Professionnels

Centre National de l'Enseignement Professionnel à Distance (CNEPD)

المركز الوطني للتعليم المهني عن بعد (م.و.ت.م.ب)

COURS D 'ALGORITHME

SÉRIE 04

TABLEAUX ET CHAINES DE CARACTÈRES

OBJECTIF PÉDAGOGIQUE : À la fin de cette série, le stagiaire doit être capable de comparer entre un tableau à une seul dimension, et un tableau de deux dimensions d'un programme algorithmique.

PLAN DE LA LECON :

I- LES TABLEAUX :

Supposons que l'on ait à résoudre le problème suivant : Calculer le nombre NVM des valeurs nulles qui existent dans un vecteur V donné ; de n composantes V_i réelles, puis calculer leur somme SOM = ΣV_i et imprimer à la fin du traitement les V_i, n, NVM et SOM.

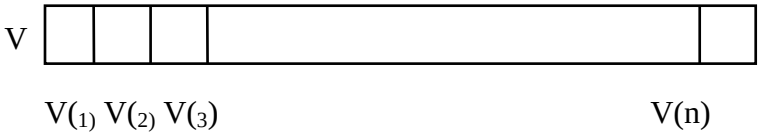
Cela suppose que l'on doit lire les valeurs V_i, les traiter pour calculer NVM et SOM sans les détruire afin de pouvoir les imprimer toutes à la fin du traitement. Il est donc nécessaire de nommer chaque V_i par une identification (V1, V2, V3...) ou bien d'avoir une structure globale qui représente V mais dans laquelle on peut distinguer V1, V2..., Vi,... Vn sans être obligé d'écraser les valeurs précédemment lues des Vi.

Cette seconde manière de faire offre l'avantage :

1) d'éviter de manipuler un grand nombre de variables qui seraient V1, V2..., Vn et qui auraient été déclarées par : V1, V2, V3, V4..., Vn: Réel ;

2) de manipuler V comme étant une seule structure (un objet composé de plusieurs objets), ce qui reflète la vraie nature du vecteur V doté d'un seul nom et de plusieurs composants.

Exemple : Soit V un vecteur à n composants, il est représenté par **:**



En assemblant des variables de même type simple (ou composé) et dont le nombre est défini à la déclaration de cette structure, nous constituons un type composé ou une structure de données que l'on appelle TABLEAU. Un tableau est donc un assemblage de variables de même type et ayant les caractéristiques suivantes :

a) Chaque composant du tableau est désigné explicitement (quand il est manipulé par les actions de l'algorithme), ce qui le rend directement accessible,

b) Le nombre de composants est défini à la déclaration du tableau et ne change pas lors du déroulement de l'algorithme.
Ces caractéristiques rendent nécessaires la définition d'un moyen de désignation des composants individuels du tableau et de définition des types de structure tableau. Ces composants sont référencés au moyen du nom de la variable du tableau et d'un indice (écrit entre parenthèse) qui désigne de façon unique l'élément désiré.

Exemple :

```
V(1) = - 1   (* le premier élément du tableau*)
V(2) =   0   (* le second élément du tableau*)
V(4) =   3   (* le quatrième élément du tableau*)
V(3) =   5   (* le troisième élément du tableau*)
```

V	-1	0	5	3

L'indice de chaque composant est un objet calculable ; autrement dit pour désigner V(1) et lui associer la valeur -56 par exemple, on associe à l'indice i la valeur 1 (par l'opération d'affectation) puis à V(i) la valeur 56.

La définition d'une variable de type tableau est donnée en algorithmique par :

<nom de variable> : TABLEAU (t1, t2..., tm) <type> ;

Où les ti désignent les dimensions maximales et <type> donne le type des composants du tableau.

Exemple :

```
V : Tableau (4) ENTIER ;     (* V est un tableau de 4 entiers *)
```

Si A est une matrice carrée n x n (avec n = 100) de réels elle est déclarée par :

```
A : TABLEAU (100, 100) REEL ;
```

II-TABLEAUX À UNE DIMENSION :

Nous avons vu qu'une variable possède un nom et un type fixe et qu'elle est associée à une valeur qui peut varier pendant l'exécution du programme.

Un tableau possède lui aussi un nom (identificateur) et un type fixe, mais il est associé à un ensemble de valeurs. Il permettra donc de représenter des ensembles de valeurs ayant des propriétés communes.

Chaque valeur est repérée par le nom du tableau suivi d'un ou de plusieurs indices. Chaque indice correspond à la dimension du tableau.
Une déclaration du tableau permet d'indiquer son type, le nombre de ses dimensions et les bornes pour chacune des indices correspondants.

Nous noterons par exemple : B : TABLEAU (1..10) ENTIER ;

On définit ainsi le tableau B qui est un tableau d'entiers à 1 dimension et pourrait contenir, par exemple, les notes d'un étudiant à dix (10) épreuves d'examen. On peut donc écrire B : TABLEAU(10) ENTIER ;

Référence à un élément d'un tableau : Considérons le cas d'un tableau à une dimension, donc à un seul indice. L'indice doit avoir une valeur entière comprise entre les bornes m et n indiquées lors de la déclaration du tableau. La référence à un élément du tableau se fait par le nom du tableau suivi d'une constante entière (indice).

Ainsi, après la déclaration B : TABLEAU (10) ENTIER ; On référencie le troisième élément du tableau par B(3).

Mais l'indice peut aussi être un identificateur de variable entière, pourvu que, à l'instant où l'on fait la référence, la valeur de cette variable soit bien comprise entre les bornes.

Exemple :

Si à un instant donné, I vaut 3 et J vaut 3 les notations B(I), B(J) et B(3) désignent à cet instant le même élément.

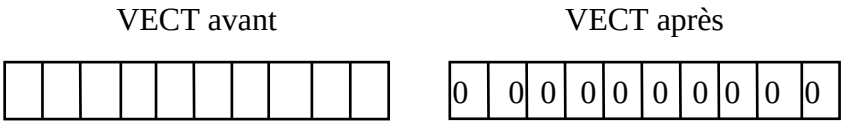
Il est essentiel de bien comprendre que la notation B(I) n'a de signification que par rapport à l' instant où la directive qui la contient est exécutée. C'est la valeur de I à cet instant qui détermine quel est l'élément du tableau concerné. La même notation B(I) dans une deuxième directive, peut donc désigner un autre élément, si I a changé de valeur entre temps.

Ces propriétés sont remarquables. Certes elles peuvent être sources d'erreurs pour un programmeur peu attentif. Mais elles offrent en revanche des possibilités logiques fondamentales par exemple en faisant varier I entre les bornes à un pas égal à 1 ; on peut adresser successivement chaque élément d'un tableau.

Exemple :

Voici un algorithme qui permet de déclarer un tableau qui a pour nom VECT, de type entier, à une dimension de 10 éléments et l'initialiser à zéro.

```
ALGORITHME TABLEAU ;
DEBUT
  I : ENTIER ;
  VECT : TABLEAU (10) ENTIER ;
  POUR     I = 1 à 10
          FAIRE
                  VECT (I) := 0 ;
          FAIT ;
 FIN.
```



Exemple :

Déclarer un tableau de nom CARRE de 100 éléments entiers, en lui insérant les carrés des nombres entiers de 1 à 100.

```
ALGORITHME CARRE ;
DEBUT
   CARRE : TABLEAU (100) ENTIER ;
   I: ENTIER;
   I: =1;
    TANTQUE I <= 100
              FAIRE   CARRE (I):= I * I;
                        ECRIRE (CARRE (I)) ;
               FAIT ;
FIN.
```

III- TABLEAU À DEUX DIMENSIONS :

De la même manière que le tableau à une dimension, on peut définir le tableau à deux dimensions comme étant un tableau avec deux indices. Lui aussi possède un identificateur on nom et un type fixe, mais il est associé à un ensemble de valeurs.

Il permettra donc de représenter des ensembles de valeurs ayant des propriétés communes, en particulier ce qu'on appelle des matrices en mathématiques. La valeur est repérée par le nom du tableau suivi de deux (02) indices qui représentent sa dimension, ainsi le premier indique la ligne et le second, quant à lui, désigne la colonne.

La définition d'une variable de type tableau à 2 dimensions s'écrit de cette manière : <nom de Var> : Tableau (t1, t2) <type> ;

Exemple :

TAB : TABLEAU (4,3) REEL ;

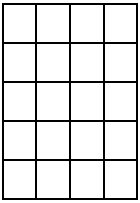
Où TAB est TABLEAU à 2 dimensions de 4 lignes et 3 colonnes, dont les éléments sont de type réel.

Exemple :

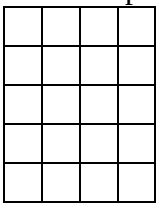
Voici un algorithme qui permet de déclarer un tableau de nom TAB à 2 dimensions constitué d'entiers en 5 lignes et 4 colonnes puis l'initialiser à zéro

```
ALGORITHME TABLEAU1 ;
DEBUT
I, J : ENTIER ;
TAB : TABLEAU (5,4) ENTIER ;
POUR I := 1 à 5
        FAIRE      POUR J := 1 à 4
                              FAIRE
                                    TAB (I, J):= 0;
                                   Ecrire TAB (I, J);
FIN.
```

TAB avant



TAB après



Exemple :

Déclarer un tableau (nommé CARRE) représentant une matrice (4,5) en lui insérant les carrés des nombres entiers à partir de 1.

```
ALGORITHME Carrés ;
DEBUT
i, j, c : ENTIER ;
CARRE : TABLEAU (4,5) ENTIER ;
C := 1 ;
Pour i := 1 à 4
        Faire
                Pour j := 1 à 5
                    Faire
                        CARRE (i, j) : = C * C ;
                        Ecrire (CARRE (i, j)) ;
                        C : =  C + 1 ;
                      Fait ;
        Fait ;
Fin.
```

IV- CHAINES DE CARACTÈRES :

Le type chaîne de caractères désigne l'ensemble qu'il est possible de former à partir de plusieurs objets de type caractère.

Une variable C de type chaîne de caractères se définie en algorithmique par : C : chaîne (n) ; notation dans laquelle n est une constante entière positive qui indique le nombre de caractères que comportera chaque valeur de C.

Notons que la valeur n défini la capacité des valeurs qui comptent exactement n caractères nous dirons que n est la longueur de la variable C.

Exemple :

PRENOM : chaîne de caractères (6) ;

L'exécution de l'action. Lire (PRENOM) ; pour une valeur tapées au clavier d'une station de travail, donnera à la variable PRENOM :

- La valeur 'FARIDA' si l'information tapée est 'FARIDA'
- LA valeur 'ALI 000' si l'information tapée est 'ALI'
- La valeur 'MOHAME' si l'information tapée est 'MOHAMED'

Nous conviendrons en effet que les troncations (les découpages des mots qui dépassent la longueur de la chaîne) se font à droite et que les valeurs trop courtes sont complétées à droite par des espaces. Ces règles sont en effet celles de la majorité des processus.

Ces chaînes sont de longueur quelconque. Elles peuvent contenir des chiffres des lettres, des blancs, des caractères spéciaux. Nous noterons entre apostrophe les constantes de type chaînes de caractères (STRING en Pascal). Ainsi 'KAMEL', 'FE023', 'X=' sont des constantes chaînes de caractères. Les variables que nous venons de décrire sont de longueur fixe : une variable de type chaîne de longueur K aura toujours pour valeur une chaîne de K

caractères. Il Peut cependant être intéressant de pouvoir disposer de variables dont les valeurs possibles peuvent varier en longueur. Montrons le à partir d'un exemple, considérons pour cela le problème illustré par la ligne ci-dessous. Soit la variable Nom Mois de type Tableau à 12 composants de type chaîne de caractères que l'on initialise par :

Nom Mois (1) : = 'JANVIER00' ;
Nom Mois (2) : = 'FEVRIER00' ;
:
:
Nom Mois (9) : = 'SEPTEMBRE' ;
Nom Mois (12) : = 'DECEMBRE0' ;

On remarque ici, qu'on complète la chaîne de caractère par des '0' qui seront omis lors du traitement, pour respecter la longueur qui est : 9.

Pour préparer la date à insérer, il suffit alors de définir la variable DATEEDITEE comme étant une variable composée de jour, mois et An :

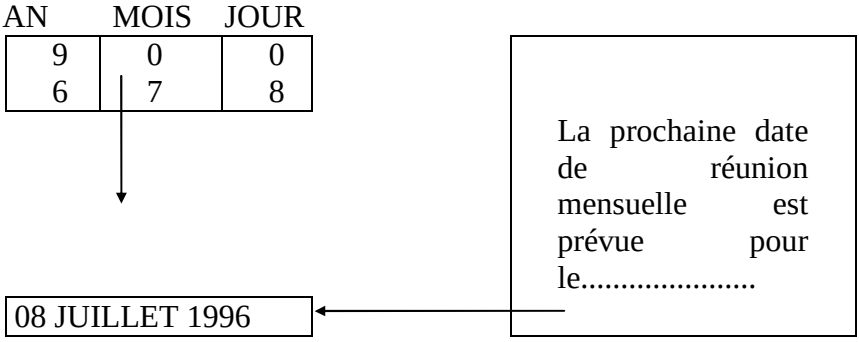
DATEEDITEE : variable composée
Jour de type Entier (1...31)
M de type chaîne de caractères (9)
An de type Entier (00...99)

Soit DATENUM une valeur lue au clavier d'une date, les affectations des champs de la variable DATEEDITEE : Jour, M, et an se font comme suit :

DATEEDITEE. Jour : = DATENUM. Jour ;
DATEEDITEE. M : = Nom Mois (DATENUM. M) ;
DATEDITEE. An : = DATENUM. An ;

Et enfin, de placer la valeur de DATEDITEE dans le texte. Ainsi avec la valeur 960796 pour DATENUM, la date à insérer sera 08 JUILLET 96.
IL s'agit d'insérer une date dans un texte sachant :

- Que cette date est donnée sous forme numérique par la valeur (année, mois, jour) d'une variable composée DATENUM.
- Que la date à insérer doit être formée du numéro du jour, du nom du mois en toutes lettres et du millésime de l'année



DATEEDITEE

La variable DATENUM peut être définie comme suit :

DATENUM : variable composée
AN de type entier ;
MOIS de type entier de 1..12 ;
JOUR de type entier de 1..31 ;

Pour préparer la date à insérer, on défini le vecteur NOMMOIS :
NOM MOIS : TABLEAU (12) chaîne de caractères ;

Exemple :

Ecrire l'algorithme qui lit puis affiche la liste des stagiaires admis à l'examen.

ALGORITHME STAGIAIRES ;

Début

NOM : CHAINE DE CARACTERES (15) ;
PRENOM : CHAINE DE CARACTERES (15) ;
NB,I : ENTIER ;
MOY : REEL ;
Ecrire ('Donner le nombre de stagiaires') ;
Lire (NB) ;
Pour I :=1 à NB
Faire Ecrire ('Donner le nom du', I,'ème stagiaire') ;
Lire (NOM) ;
Ecrire ('Donner le prénom du',I,'ème stagiaire') ;
Lire (PRENOM) ;
Ecrire ('Donner sa moyenne') ;
Lire (MOY) ;
Si MOY > 10
Alors Ecrire (NOM, PRENOM, 'est un stagiaire admis') ;
Fsi ;
Fait ;
Fin.

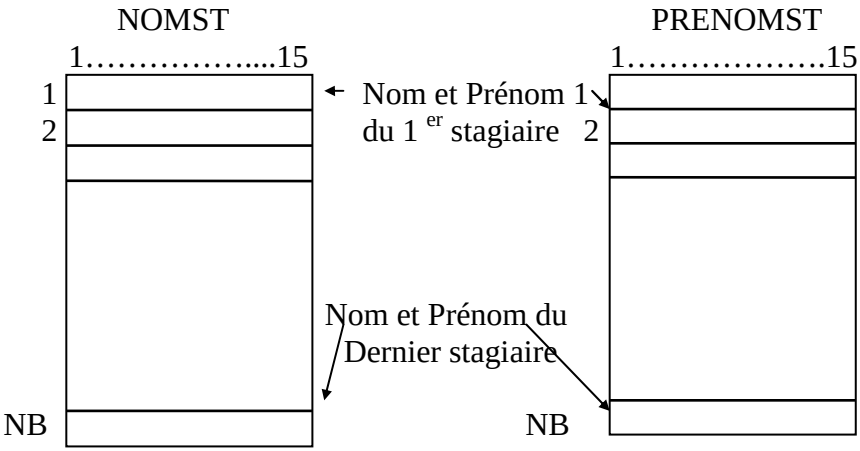
Nous remarquons que cet algorithme écrase à chaque fois les valeurs des variables NOM, PRENOM et MOY.

Supposons que nous voulons conserver les renseignements de tous les stagiaires saisis pour d'éventuels traitements. Nous devons donc utiliser des structures composées de plusieurs chaînes de caractères. On déclare ainsi un tableau composé de chaînes de caractères. L'algorithme devient alors :

```
ALGORITHME STAGIAIRES2 ;

Début
NOM : CHAINE DE CARACTERES (15) ;
PRENOM : CHAINE DE CARACTERES (15) ;
NOMST : TABLEAU (100) NOM ; (* NOM et PRENOM sont
des types*)
PRENOMST : TABLEAU (100) PRENOM ;(*définis au début
de *)
MOY : TABLEAU (100) REEL ; (* l'algorithme*)
NB,I: ENTIER ;
Ecrire ('Donner le nombre de stagiaires') ;
Lire (NB) ;
Pour I :=1 à NB
 Faire     Ecrire ('Donner le nom du', I,'ème  stagiaire') ;
                Lire (NOMST (I)) ;
                Ecrire ('Donner le prénom du',I,'ème  stagiaire') ;
                Lire (PRENOMST (I)) ;
                Ecrire ('Donner sa moyenne') ;
                Lire (MOY (I)) ;
                Si  MOY (I) > 10
                   Alors  Ecrire (NOMST(I), PRENOMST(I), 'est
admis') ;
                Fsi ;
        Fait ;
Fin.
```

Ici, NOMST et PRENOMST se représentent comme suit :



RÉSUMÉ :

Après les types de données simples, cette leçon a fait l'objet de deux types composés qui sont respectivement le type TABLEAU et le type CHAINE DE CARACTERES.

Se sont deux types structurés de plusieurs éléments de même type et dont la longueur est définie au préalable. Donc un tableau est un ensemble d'éléments qui peuvent être de type ENTIER, REEL, CARACTERE ou BOOLEEN par contre une chaîne de caractères est un ensemble d'éléments de type CARACTERE.

On accède aux éléments d'un tableau un par un , on associant à l'identificateur du tableau une variable entre parenthèse appelée indice qui prend les valeurs respectivement 1, 2, 3,....., la taille maximale du tableau, pour référencer respectivement la première, deuxième, troisième,.... et la dernière composante du tableau et on accède à la chaîne de caractères par son identificateur en référençant toutes ses composantes à la fois .

Nous avons vu, le tableau à une dimension pour lequel un seul indice permet d'accéder à ses éléments et le tableau à deux dimensions pour lequel deux indices sont nécessaires pour accéder à ses éléments.

EXERCICES D'APPLICATION :

EXERCICE N°01 :

Ecrire l'algorithme qui recherche le nombre maximum et le nombre minimum d'une suite de N entiers donnés. (N<=100)

EXERCICE N°02 :

On dispose de deux suites d'entiers positifs ou nuls :

$X = x_1, x_2, ..., x_n$ $\quad n < 100$
$Y = y_1, y_2, ..., y_n$ $\quad m < 150$

On s'intéresse à l'ensemble des couples formés d'un élément de X et d'un élément de Y et on associe un poids à chacun des couples, défini de la manière suivante :
POIDS $(x_i, y_i) = | x_i - y_i |$

Ecrire l'algorithme qui permet de trouver le couple de plus petit poids.

Notons que : $|x_i - y_i| = \begin{cases} x_i - y_i & si \quad x_i >= y_i \\ y_i - x_i & si \quad y_i > x_i \end{cases}$

EXERCICE N°03 :

Ecrire l'algorithme qui lit à partir du clavier une suite finie de mots et n'affiche que celui qui correspond à « ALGERIE ».

EXERCICE N°04 :

Soient deux matrices carrées (n, n) de nombres réels. Ecrire l'algorithme qui calcule leur somme.

EXERCICE N°05 :

Ecrire l'algorithme qui lit la liste des trois notes obtenues dans un examen, calcule leur moyenne puis l'affiche en conservant toutes ces données.

CORRECTION DES EXERCICES :

EXERCICE N°01 :

```
ALGORITHME MIN-MAX ;
Début
S : TABLEAU (100) ENTIER ;
I, N, MAX, MIN : ENTIER ;

Ecrire (‘Donner la valeur de N’) ;
Lire (N) ;
Pour I := 1 à N
      Faire
            Ecrire (‘Donner le’, I, ‘ère élément de la suite’) ;
            Lire (S(I)) ;
      Fait ;
MAX := MININT ;  (* MININT est la valeur de la plus petite constante *)
MIN := MAXINT ;  (* MAXINT est la valeur de la plus grande constante *)
Pour I := 1 à N
      Faire
            Si S(I) > MAX
                  Alors MAX: = S (I);
              Fsi;
            Si S (I) < MIN
                  Alors MIN: = S (I);
            Fsi ;
       Fait ;
Ecrire (‘Le maximum de la suite est :’, MAX) ;
Ecrire (‘Le minimum de la suite est :’, MIN) ;
Fin.
```

EXERCICE N°02 :

```
ALGORITHME POIDS ;
Début
X : TABLEAU (100) ENTIER ;
Y : TABLEAU (150)  ENTIER ;
MIN, I, M, N, POIDS, PPOIDS : ENTIER ;
Ecrire ('Donner la longueur de la suite X et celle de Y') ;
Lire (M , N ) ;
Pour I := 1 à M
      Faire
            Ecrire ('Donner le', I, ' eme élément de X' ) ;
             Lire (X(I)) ;
      Fait ;
Pour I := 1 à N
      Faire
            Ecrire ('Donner le', I, ' eme élément de Y' ) ;
             Lire (Y(I)) ;
      Fait ;
Si N < M
    Alors   MIN := N
    Sinon  MIN := M ;
Fsi ;
Si  X(1) >= Y(1)
   Alors   PPOIDS := X(1) - Y(1)
   Sinon   PPOIDS := Y(1) - X(1) ;
Fsi ;
Pour  I := 2  à   MIN
       Faire   Si  X(I) >= Y(I)
                 Alors    POIDS: = X (I) - Y (I)
                    Sinon   POIDS: = Y (I) - X (I) ;
               Fsi ;
               Si  POIDS < PPOIDS
                   Alors    PPOIDS := POIDS ;
               Fsi ;
       Fait ;
Ecrire (' Le plus petit poids est :', PPOIDS) ;
Fin.
```

EXERCICE N°03 :

```
ALGORITHME MOT ;
Début
MOT : CHAINE DE CARACTERES (10) ;
I, N : ENTIER ;
Ecrire ('Donner le nombre de mots') ;

Lire (N) ;
Pour    I := 1     à   N
        Faire   Lire (MOT) ;
                Si  MOT =  « ALGERIE »
                    Alors  Ecrire (MOT) ;
                 Fsi ;
        Fait ;
Fin.
```

EXERCICE N°04 :

```
ALGORITHME SOMME-MATRICE ;
Début
A, B, C : TABLEAU (100) REEL ;
I, J, N : ENTIER ;
Ecrire ('Donner la valeur de N') ;
Lire (N) ;
Pour  I := 1   à  N
        Faire   Pour  J := 1  à  N
                        Faire Lire (A (I, J), B (I, J));
                                C (I, J):= A (I, J) + B (I, J) ;
                         Fait;
        Fait ;
Ecrire ('La matrice somme est :') ;
Pour  I := 1   à  N
        Faire    Pour  J := 1   à  N
                            Faire   Ecrire (C (I,J) ) ;
                            Fait ;
             Fait ;
Fin.
```

EXERCICE N°05 :

```
ALGORITHME MOYENNE ;
Début
NOM : CHAINE DE CARACTERES (15) ;
PRENOM : CHAINE DE CARACTERES (15) ;
LISTE-N : TABLEAU (200) NOM ;
LISTE-P: TABLEAU (200) PRENOM ;


NOTE1: TABLEAU (200) REEL ;
NOTE2: TABLEAU (200) REEL ;
NOTE3: TABLEAU (200) REEL ;
MOY: TABLEAU (200) REEL ;
I, N : ENTIER ;
Ecrire (‘Donner le nombre de candidats’) ;
Lire (N) ;
Pour  I := 1  à   N
       Faire  Ecrire (‘Donner le nom :’) ;
                 Lire (LISTE-N (I)) ;
                 Ecrire (‘Donner le prénom :’) ;
                 Lire (LISTE-P (I)) ;
                 Ecrire (‘Donner la première note :’) ;
                 Lire (NOTE1 (I)) ;
                 Ecrire (‘Donner la deuxième note :’) ;
                 Lire (NOTE2 (I)) ;
                 Ecrire (‘Donner la troisième note :’) ;
                 Lire (NOTE3 (I));
                 MOY  (I):= (NOTE1  (I)  +  NOTE2  (I)  +  NOTE3
(I)/ 3;
        Fait ;
Pour  I := 1  à  N
Faire

Écrire  (LISTE-N  (I),  LISTE-P  (I),  NOTE1  (I),  NOTE2  (I),
NOTE3 (I), MOY (I));
Fait ;
Fin.
```